Introduction to Linux Basics

Part-I

Georgia Advanced Computing Resource Center
University of Georgia
Zhuofei Hou, HPC Trainer
zhuofei@uga.edu
Outline

• What is GACRC?
• What is Linux?
• Linux Command, Shell and Filesystem Concepts
• Linux Common Commands
What is GACRC?

Who Are We?
- Georgia Advanced Computing Resource Center
- Collaboration between the Office of Vice President for Research (OVPR) and the Office of the Vice President for Information Technology (OVPIT)
- Guided by a faculty advisory committee (GACRC-AC)

Why Are We Here?
- To provide computing hardware and network infrastructure in support of high-performance computing (HPC) at UGA

Where Are We?
- [http://gacrc.uga.edu](http://gacrc.uga.edu) (Web) [http://wiki.gacrc.uga.edu](http://wiki.gacrc.uga.edu) (Wiki)
- [https://wiki.gacrc.uga.edu/wiki/Getting_Help](https://wiki.gacrc.uga.edu/wiki/Getting_Help) (Support)
- [https://blog.gacrc.uga.edu](https://blog.gacrc.uga.edu) (Blog) [http://forums.gacrc.uga.edu](http://forums.gacrc.uga.edu) (Forums)
What is Linux?

- What is Operating System (OS)?
- What is Linux OS?
- Brief History of Linux OS
- Why Linux OS?
What is Linux – Operating System

- Operating System (OS):
  - Program initially loaded at booting time, to manage all the other application programs on a computer
  - Software interface between computer hardware and its human user

- Needed for ALL computers to be operated
- Needed to run software and control hardware

- Popular OSes:
  - Windows
  - Linux
  - macOS
What is Linux – Linux OS

- Linux OS is a full-fledged OS with 4 major parts:
  I. **Kernel**: Low-level OS, handling files, disks, RAM, networking, etc.
  II. **Supplied Programs**: Web browsing, Audio, Video, DVD burning......
  III. **The Shell**: A command-line user interface for a user to type and execute commands:
    - Bourne Shell (sh)
    - Korn Shell (ksh)  
    - C Shell (csh)  
    - Bourne-Again Shell (bash)  →  Linux default shell
  IV. **X**: A graphical system providing graphical user interface (GUI)
What is Linux OS – Brief History

- Originally was a kernel only, nothing else
- Combined with the various software and compilers from GNU Project to form an OS, called as GNU/Linux OS:

  \[
  \text{Linux Kernel } + \text{ GNU Components } \rightarrow \text{ GNU/Linux OS } \rightarrow \text{ Linux OS}
  \]

- So, History of Linux = History of Linux Kernel + History of GNU
What is Linux OS – Brief History of Linux Kernel

- Developed in 1991 by Linus Torvalds, a second year student, at the University of Helsinki, Finland
- Developed as a clone of UNIX OS, which is cheaper, can run on PC, and is nonproprietary
- Linux 0.02 released in 1991 consists of only the kernel and 3 utilities:
  - Bash: a command-line interface (CLI)
  - update: a utility to flush file system buffers
  - gcc: a C++ compiler
What is Linux OS – Brief History of GNU Project

- Started in 1983 by Richard Stallman. Launched in 1984 with a mission to develop a complete UNIX-like OS which is FREE for copying and modification.
- GNU means “GNU's Not Unix”
- However, NO functional kernel developed by GNU itself …..
- Linux kernel was the BEST fit as the kernel for the GNU Project, SO …..
What is Linux OS – Brief History

- Today, Linux OS is used by millions and available in the form of various Linux distributions:

- Linux is the most used OS on servers:
  - As of February 2010, 6 out of 10 most reliable web hosting companies
  - As of November 2014, 485 (97%) out of top 500 supercomputers

- Linux OS is supported by many big companies, such as IBM, Google, Sun, Novell, Oracle, HP, Dell, etc.

(Data are cited from http://en.wikipedia.org/wiki/Linux)
What is Linux OS – Why Linux?

• Viruses FREE
• Very STABLE
• FREE Linux OS
• Never gets slow
• No need to defrag hard disk
• Highest degree to customize user’s working environment
• Comes with most of the required software pre-installed
• Update all software with minimum labor
Linux Command, Shell and Filesystem Concepts

- What is a Command?
- What is a Shell?
- What is Filesystem?
**Linux Command, Shell and Filesystem**

What’s a Command ➜ A Linux command typically consists of a *program* followed by *command options* and *arguments*, typed within a *shell*:

```
$ wc -l -w file1
```

**OR**

```
$ wc -lw file1 file2
```

**Output:**

```
15    86     file1
15    86   file1
100  2104   file2
115  2190   total
```

# of lines   # of words
Linux Command, Shell and Filesystem

What’s a Command ➔ A Linux command typically consists of a *program* followed by *command options* and *arguments*, typed within a *shell*:

- **3 general formats of command options:**
  
  i. *with no value*: `wc -l -w`
  
  ii. *with a value*: `blastx -thread 4`
  
  iii. *combined*: `wc -lw`

- **5 Tips:**
  
  i. Linux command is ALWAYS *case sensitive*!
  
  ii. Press **TAB** key to autocomplete a command or filename ➔ Auto-completion
  
  iii. Press ↑ and ↓ arrow keys to look up previous commands ➔ Command history
  
  iv. Press **CTRL+c** to terminate a command
  
  v. How to use a command? Use command option **--help**, e.g., `wc --help`

  Use **man** command, e.g., `man wc`
Linux Command, Shell and Filesystem

- **What’s a Shell** ➔ A place to type and run commands on a Linux system:
  - **Command-line user interface** for typing commands
  - **Command interpreter** to interpret & run commands
  - **Programming environment** for scripting

- **Linux default:** **Bourne-Again Shell (bash)**

- **To open a shell on:**

<table>
<thead>
<tr>
<th>Location</th>
<th>Shell Window</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Linux/Mac</td>
<td>shell window</td>
<td>Terminal</td>
</tr>
<tr>
<td>Local windows</td>
<td>shell window</td>
<td>Cygwin</td>
</tr>
<tr>
<td>Remote Linux machine</td>
<td>a shell will run immediately when log in</td>
<td></td>
</tr>
</tbody>
</table>
Linux Command, Shell and Filesystem

- What’s Filesystem ➔ A internal data structure that OS uses to organize files on disk:
  - Tree-structured & hierarchical
  - Topmost directory: root directory (/)
  - Each directory has one parent (except for /), may contain 0 or more subdirectories
  - Files are collected in directories
  - Files and directories are accessed by path:
    - path 1: /home/mkl/mystuff/stuff
    - path 2: /usr/local/
  - A path beginning with /: an absolute path
Linux Command, Shell and Filesystem

- What’s Filesystem ➫ A internal data structure that OS uses to organize files on disk:
  - Two special directories:
    - . (a single dot) : your current directory
    - .. (two dots in a row) : parent directory
  - E.g. If current directory is /home/mkl
    - path 1: go . = go /home/mkl
    - path 2: go .. = go /home
    - path 3: go ../other = go /home/other
    - path 4: go ../../usr = go /usr
  - ✓ A path not beginning with /: a relative path
Linux Command, Shell and Filesystem

What’s Filesystem ➔ A internal data structure that OS uses to organize files on disk:

✔ Filename naming convention:
  i. Good characters: A ~ Z or a ~ z, 0 ~ 9, _ (Underscore), . (Period), - (Dash)
  ii. Bad characters: special characters, e.g., $, *, ?, /, |, #, &, <, > and whitespace
  iii. Linux filename is ALWAYS case sensitive!
  iv. Not like Windows, no file extension needed in Linux!
  v. Max length of a filename is usually 255 characters

✔ Examples:
  i. Good: matrixdata1, matric_data_1, matrix.data.1, _testFile, 20150720, etc.
  ii. Bad: xy*z, x>y, $myfile, matrix|data, datafile&, matrix data, etc.
Linux Common Commands

- Basic File Operations
- Directory Operations
- File Viewing
- Other

(For more complete list, please refer to GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Command_List)

Please do NOT do command practice on Login node of GACRC clusters!
Linux Common Commands – Basic File Operations

- `ls`: List files and subdirectories in a directory
- `cp`: Copy a file into another or a directory
- `mv`: Rename or move a file into a directory
- `rm`: Remove a file ⚠️
# Linux Common Commands – Basic File Operations

- **ls**: List files and subdirectories in a directory
  - `ls -l`: List files with a long information listing
  - `ls -a`: List all files, including *hidden configuration files*, whose names begin with a dot, called as “*dot files*”
  - `ls -h`: List files with sizes in human readable format
  - `ls -lh`: Combination of `-l` and `-h`

```
$ ls -lha
-rw-r--r-- 1 jsmith abclab  336 Jul 16 10:06 .bashrc  ➔ dot file
drwxr-xr-x 2 jsmith abclab 4.0K Jul 16 10:05 data       ➔ subdirectory
-rw------- 1 jsmith abclab 402 Jul 16 10:05 hello1.c    ➔ C source code
-rw-r-xr-x 1 jsmith abclab 188K Jul 16 10:05 hello1.x    ➔ C binary
-rw-r--r-- 1 jsmith abclab 252 Jul 16 10:05 README      ➔ readme file
-rw-r--r-- 1 jsmith abclab 131 Jul 16 10:05 sub1.sh      ➔ shell script
```
# Linux Common Commands – Basic File Operations

<table>
<thead>
<tr>
<th>File Type</th>
<th># of Hard Links</th>
<th>Permissions</th>
<th>Owners</th>
<th>File Size</th>
<th>Last Modification Time</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>-rwxr-xr-x</td>
<td>1</td>
<td>jsmith</td>
<td>abclab</td>
<td>188K</td>
<td>Jul 16 10:05</td>
<td>hello1.x</td>
</tr>
</tbody>
</table>

**File Type**: `-` for Regular File  
**d** for Directory  
**File Permissions**:  
- **r** for Read (4)  
- **w** for Write (2)  
- **x** for Execute (1)  

**User**: (r+w+x)=7  
**Group**: (r+x)=5  
**Other**: (r+x)=5  

`hello1.x` is a regular file with permissions of 755  

To change permissions: `chmod` command
Linux Common Commands – Basic File Operations

- **`cp`**: Copy a file into another or a directory

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cp file1 file2</code></td>
<td>Copy a file into another</td>
<td>zcluster$ cp hello1.c hello2.c (\Rightarrow) hello2.c is a new file copied from hello1.c</td>
</tr>
<tr>
<td><code>cp file directory</code></td>
<td>Copy a file into a directory</td>
<td>zcluster$ cp hello1.c ./data (\Rightarrow) ./data is a subdirectory</td>
</tr>
<tr>
<td><code>cp -i file1 file2</code></td>
<td>Copy with interactive mode, ask before overwriting</td>
<td>zcluster$ cp -i hello1.c hello2.c (\Rightarrow) interactive mode is always safe!</td>
</tr>
</tbody>
</table>

- `cp`: overwrite `hello2.c`? n
# Linux Common Commands – Basic File Operations

- **mv**: Rename or move a file into a directory

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mv file1 file2</code></td>
<td>Rename a file</td>
</tr>
<tr>
<td><code>mv file directory</code></td>
<td>Move a file into a directory</td>
</tr>
<tr>
<td><code>mv -i file1 file2</code></td>
<td>Move with interactive mode, ask before overwriting</td>
</tr>
</tbody>
</table>

- Examples:
  - `zcluster$ mv hello1.c ./data` ➡️ `./data` is a subdirectory
  - `zcluster$ mv hello1.c hello2.c` ➡️ `hello2.c` is the file renamed from `hello1.c`
  - `zcluster$ mv hello1.c ./data` ➡️ `./data` is a subdirectory
  - `zcluster$ mv -i hello1.c hello2.c` ➡️ `interactive mode is always safe!`
  - `mv: overwrite `hello2.c'? n`

  `zcluster$`
Linux Common Commands – Basic File Operations

- **rm**: Remove a file

  - `rm file`  
    Remove a file
  - `rm -i file`  
    Remove with interactive mode, ask before deleting a file

```
zcluster$ rm hello2.c  
  ➔ hello2.c is removed from current directory

zcluster$ rm -i hello2.c  
  ➔ interactive mode is always safe!
```

- `rm`: Remove a file

  - `rm file`  
    Remove a file
  - `rm -i file`  
    Remove with interactive mode, ask before deleting a file

```
zcluster$ rm hello2.c  
  ➔ hello2.c is removed from current directory

zcluster$ rm -i hello2.c  
  ➔ interactive mode is always safe!
```
Linux Common Commands – Directory Operations

- **cd**: Change your current working directory
- **pwd**: Print absolute path of your current working directory
- **mkdir**: Create a directory
- **rmdir**: Delete an empty directory
- **rm -r**: Delete a nonempty directory and its contents
Linux Common Commands – Directory Operations

- **cd**: Change your current working directory

  ```
  cd dirname
  ```
  Change to the dirname directory

  - `zcluster$ cd ./date`
    - change to a subdirectory `.data`
  - `zcluster$ cd ..`
    - change to parent directory
  - `zcluster$ cd ~/test`
    - change to a subdirectory `./test` in home directory (~)
  - `zcluster$ cd /home/abclab/jsmith/test`
    - an absolute path is used!

- **pwd**: Print absolute path of your current working directory

  ```
  pwd
  ```
  /home/abclab/jsmith/test/data

  - an absolute path is printed!
Linux Common Commands – Directory Operations

- **mkdir**: Create a directory
  
  ```
  mkdir dirname
  ```
  Make a directory with the name of dirname

  ```
  zcluster$ mkdir data1
  ```
  Create a subdirectory in current working directory

- **rmdir**: Delete an empty directory
  
  ```
  rmdir dirname
  ```
  Remove an empty directory

  ```
  zcluster$ rmdir data1
  ```
  data1 is an empty directory!

- **rm -r**: Delete a nonempty directory and its contents
  
  ```
  rmdir -ri dirname
  ```
  Remove with interactive mode, ask before removing

  ```
  zcluster$ rm -ri data1
  ```
  interactive mode is always safe!
Linux Common Commands – File Viewing

- **cat**: Print files to standard output, concatenating them
- **less**: View text files, one screen at a time, scroll down/up
- **more**: View text files, one screen at a time, scroll down only
# Linux Common Commands – File Viewing

- **cat**: Print files to standard output, concatenating them

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cat file</code></td>
<td>Print contents of file1 to standard output</td>
</tr>
<tr>
<td><code>cat file1 file2</code></td>
<td>Print contents of files to standard output, concatenating them</td>
</tr>
</tbody>
</table>

```
zcluster$ cat file1
Hello, this is file1.
zcluster$ cat file2
Hello, this is file2.
zcluster $ cat file1 file2
Hello, this is file1. Hello, this is file2.
```

- ➡️ print contents of **file1**
- ➡️ print contents of **file2**
- ➡️ print contents of **file1** and **file2** with concatenation
Linux Common Commands – File Viewing

- **less**: View text files, one screen at a time, scroll down and up
  
  ```
  less file
  ```
  View text one “page” at a time, `spacebar` to scroll down, key `b` to scroll up, key `q` to quit

  ```
  zcluster$ less file1
  ```

- **more**: View text files, one screen at a time, scroll down only
  
  ```
  more file
  ```
  View text one “page” at a time, `spacebar` to scroll down,

  ```
  zcluster$ more file1
  ```
Linux Common Commands – Other

- `file` : Determine the type of a file
- `dos2unix` : Convert DOS/Windows file to Linux format
- `mac2unix` : Convert Mac file to Linux format
Linux Common Commands – Other

- **file**: Report the type of a file

  ```
  file file1
  ```

  Report the type of the file file1

  - `zcluster$ file data`
    - `data`: directory
  - `zcluster$ file hello1.c`
    - `hello1.c`: ASCII C program text
  - `zcluster$ file hello1.x`
    - `hello1.x`: ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared libs), not stripped
  - `zcluster$ file README`
    - `README`: ASCII text
  - `zcluster$ file sub1.sh`
    - `sub1.sh`: Bourne-Again shell script text executable

  ```
  ➡ directory ./data
  ➡ programming language source file hello1.c
  ➡ executable file hello1.x
  ➡ ASCII text file README
  ➡ shell script sub1.sh
  ```
Linux Common Commands – Other

- **dos2unix**: Convert DOS/Windows file to Linux format

  ```
  dos2unix file1
  ```

  Removes DOS/Windows line endings in file1

  ```
  zcluster$ dos2unix file1
  ```

- **mac2unix**: Convert Mac file to Linux format

  ```
  mac2unix file1
  ```

  Removes Mac line endings in file1

  ```
  zcluster$ mac2unix file1
  ```
Thank You!